НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

Завідувач кафедри

_____ О.В. Коваль
(підпис) (ініціали, прізвище)

“___”_____2019р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напряму підготовки 6.050103 “ Програмна інженерія “

на тему “Web-ресурс для організації та проведення тестів і опитувань у навчальному процесі”

Виконав (-ла): студент (-ка) 4 курсу, групи ТІ-51

Мельник Матвій Олегович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник доцент, к.т.н. Карпенко Євген Юрійович _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент_____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший , бакалаврський

Напрям підготовки 6.050103 “ Програмна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.В. Коваль
(підпис)
”___”_____2019р.

ЗАВДАННЯ
на дипломну роботу студенту
Мельнику Матвію Олеговичу
(прізвище, ім’я, по батькові)

1. Тема роботи “Web-ресурс для організації та проведення тестів і опитувань у навчальному процесі”

керівник роботи доцент, к.т.н. Карпенко Євген Юрійович
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від 22.05.2019р. № 1325-с

2. Строк подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи мови програмування JavaScript, PHP, веб-сервер Apache, система керування базами даних MySQL.

4.Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити) проаналізувати існуючі програмні рішення, обґрунтувати обрані програмні застосунки та шляхи розробки програмної системи, розробити веб-ресурс для організації та проведення тестів і опитувань у навчальному процесі, зробити висновки за результатами роботи.

5. Перелік ілюстративного матеріалу
1. Функції розробленої системи. 2. Архітектура програмної системи. 3. Концептуальна модель бази даних. 4. Головна сторінка. 5. Пункт меню “Тести”. 6. Пункт меню “Користувачі”. 7. Створення тесту. 8. Проходження тесту.

6. Дата видачі завдання 10 жовтня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі		
2	Розробка архітектури та загальної структури системи		
3.	Розробка структур окремих підсистем		
4.	Програмна реалізація системи		
5.	Оформлення пояснювальної записки		
6.	Захист програмного продукту		
7.	Передзахист		
8.	Захист		

Студент _____ Мельник М. О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Карпенко Є. Ю.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Мета роботи — розробка Web-ресурсу для організації та проведення тестів і опитувань у навчальному процесі. Для створення ресурсу використано мови програмування JavaScript, PHP, веб-сервер Apache, систему керування базами даних MySQL, технологію AJAX.

Розроблений програмний продукт дозволяє переглядати дані пройдених тестів і опитувань, а також створювати нові або проходити їх.

Записка містить 71 сторінку, 13 рисунків, 9 таблиць, 3 додатки і 24 посилання.

Ключові слова: Web-ресурс, тести, опитування, навчальний процес.

ABSTRACT

The purpose of the work is to develop a Web resource for organizing and conducting tests and surveys in the learning process. Programming languages JavaScript, PHP, Apache web server, MySQL Database Management System, AJAX technology were used for creating a resource.

The developed software allows to view data about passed tests and surveys, as well as create new ones or pass them.

The note contains 71 pages, 13 pictures, 9 tables, 3 attachments and 24 links.

Keywords: web resource, tests, surveys, learning process.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СУБД — система управління базами даних.

HTML — Hypertext Markup Language.

API — Application Programming Interface.

HTTP — Hypertext Transfer Protocol.

XML — Extensible Markup Language.

JSON — JavaScript Object Notation.

SOAP — Simple Object Access Protocol.

SOA — Service-oriented Architecture.

REST — Representational State Transfer.

ROA — Resource-oriented Architecture.

RDF — Resource Description Framework.

MVVM — Model-View-ViewModel.

MVC — Model-View-Controller.

AJAX — Asynchronous JavaScript and XML.

DHTML — Dynamic HTML.

DOM — Document Object Model.

CSS — Cascading Style Sheets.

URL — Uniform Resource Locator.

FTP — File Transfer Protocol.

ПЗ — програмне забезпечення.

UID — User identifier.

GID — Group identifier.

SQL — Structured Query Language.

ВСТУП

Останні роки з розвитком сучасних технологій, поширенням і доступністю Інтернет-зв’язку відкриваються унікальні можливості для освіти. Інтернет є джерелом активної інтелектуальної діяльності сучасного студента, який має необмежені можливості для одержання знань, удосконалення умінь, навичок. Викладачеві, у свою чергу, надається можливість оптимізувати систему контролю, зокрема переводячи тестування в режим онлайн. Значний інтерес до створення та використання освітніх веб-ресурсів обумовлений серйозними, і в значній мірі, обґрунтованими очікуваннями підвищення ефективності навчального процесу і якості навчання.

Освітні веб-ресурси — це інформаційні дані, які відображають певну предметну галузь освіти та призначені для забезпечення процесу навчання особистості, формування її знань, умінь та навичок.

Використання постійного контролю знань у навчальному процесі значно підвищує мотивацію навчання, однак для проведення тестування викладачі мають виконати велику кількість роботи для формування завдань та контролю за їх виконанням. Також під час навчання для студентів можуть проводитись опитування з метою покращення якості освітнього процесу, для цього викладачам також треба виконати значну кількість роботи, щоб підготувати всі необхідні питання та організувати сам процес.

Тому, було запропоновано дослідити можливість створення веб-ресурсу для організації та проведення тестів і опитувань, що дозволить дуже швидко здійснювати контроль, максимально автоматизувати процеси тестування і опитування, а також використовувати тести і опитування у дистанційному навчанні.

Засобами розробки є сучасні методи динамічної побудови веб-сторінок за допомогою мов програмування JavaScript, PHP, СУБД MySQL.

Пояснювальна записка представлена п’ятьма розділами. В першому розділі

сформульовано постановку задачі й розглянуто проблеми щодо організації та проведення тестів і опитувань; у другому розділі проаналізовано існуючі технології організації та проведення тестів і опитувань; у третьому розділі проводиться обґрунтування прикладних засобів програмування поставленої задачі; в четвертому розділі проводиться опис програмної реалізації, зокрема інформація про складові компоненти програмного забезпечення та їх взаємодію; у п’ятому розділі описується методика роботи користувача з програмною системою, детальний опис роботи з програмним додатком.

1. ЗАДАЧА РОЗРОБКИ WEB-РЕСУРСУ ДЛЯ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТЕСТІВ І ОПИТУВАНЬ У НАВЧАЛЬНОМУ ПРОЦЕСІ

У сучасному світі за останні роки з розвитком сучасних технологій, поширенням і доступністю Інтернет-зв'язку відкриваються унікальні можливості для освіти. Інтернет — це не лише невичерпна скарбниця освітньої інформації, а й джерело активної інтелектуальної діяльності сучасного студента, який має необмежені можливості для одержання знань, удосконалення умінь, навичок. Викладачеві, у свою чергу, надається можливість оптимізувати систему контролю, зокрема переводячи тестування в режим онлайн. Значний інтерес до створення та використання освітніх веб-ресурсів обумовлений серйозними, і в значній мірі, обґрунтованими очікуваннями підвищення ефективності навчального процесу і якості навчання.

Задачі, які повинні вирішуватись:

— забезпечення користувача персональним кабінетом, в якому можна створювати тести і опитування;

— забезпечення дуже швидкого здійснення контролю знань та опитування;

— надання можливості максимально автоматизувати процеси тестування та опитування, використовувати тести і опитування у дистанційному навчанні;

— забезпечення користувача зручним графічним інтерфейсом, який може працювати як на комп'ютері, так і на мобільному телефоні.

Вхідними даними мають бути: інформація про тести та опитування, що зберігається у базі даних.

Вихідними даними мають бути: HTML-сторінка з відображенням інформації про тест чи опитування.

Результатами роботи мають бути: результати пройдених тестів та опитувань.

Для зручності користування мають бути: розроблений пошук, зручність

представлення інформації (наглядне меню, короткий огляд інформації по тестах чи опитуваннях, з можливістю перегляду повних результатів).

Програмне забезпечення має бути з декількома програмними засобами реалізації, архітектура програмної системи має складатися з трьох компонентів: клієнт, сервер і база даних.

Користувачами системи мають бути:

— адміністратор системи;

— викладач;

— студент.

Адміністратор системи повинен мати можливість реєструвати у системі викладачів та студентів, редагувати інформацію про них, при необхідності видаляти застарілу інформацію.

Викладачі повинні мати можливість створювати тести і опитування, а також переглядати результати їх проходження студентами.

Студенти повинні мати можливість знайти потрібний тест чи опитування для його проходження, а також переглядати результати пройдених тестів і опитувань.

Система повинна надавати повну інформацію про тести і опитування, створені викладачами для студентів. Ресурс повинен бути швидким у інтерпретації найпоширенішими браузерами.

2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТЕСТІВ І ОПИТУВАНЬ У НАВЧАЛЬНОМУ ПРОЦЕСІ

2.1. Існуючі програмні засоби організації та проведення тестів і опитувань у навчальному процесі

З кожним роком набувають все більшого поширення різноманітні веб-ресурси. На їх основі вже побудовано багато різноманітного програмного забезпечення, яке вирішує дуже широкий спектр прикладних задач. Серед них можна виділити і задачу організації та проведення тестів і опитувань у навчальному процесі.

У процесі пошуку інформації, та аналізу існуючих рішень, було виявлено, що на даний момент такі системи є досить складними у використанні, або реалізують поставлену задачу не в повному обсязі:

— Освітній веб-ресурс “Perfect English Grammar” являє собою довідник, а також збірник різноманітних вправ та онлайн-тестів з граматики англійської мови. Недоліком даної системи для поставленої задачі можна вважати, що вона не має можливості авторизуватися у системі, оскільки авторизація дає змогу зберегти отримані результати;

— Веб-ресурс “Quizlet” — це ресурс для вивчення іноземних мов, у якому користувачі мають змогу тренуватися, виконуючи різні завдання та тести. Дане програмне забезпечення не реалізує поставлену задачу в повному обсязі, оскільки цей ресурс використовується виключно для підготовки до іспитів, а не для їх проходження.

Також недоліком цих систем є їх громіздкість. Для початківців ці програми можуть здатися переповненими функціоналом та дуже специфічними в плані налаштування.

Так як автоматизація проведення тестів у навчальному процесі набуває

широкого розповсюдження, то розробка зручної, інтуїтивно зрозумілої, гнучкої та невимогливої до апаратного забезпечення системи для проведення тестів і опитувань є актуальною задачею.

Одним із варіантів, який зможе працювати на будь-яких сучасних пристроях, є розробка програмної системи, яка заснована на веб-технологіях.

2.2. Поняття веб-ресурсів для організації тестів і опитувань у навчальному процесі

Ключові слова: веб-ресурс, навчальний процес, електронні освітні ресурси, освітні веб-ресурси, тестування, опитування.

Веб-ресурс — це сукупність інтегрованих засобів технічного і програмно-апаратного характеру, а також інформації, призначеної для публікації у Всесвітній мережі.

Навчальний процес — це система організаційних і дидактичних заходів, спрямованих на реалізацію змісту освіти на певному освітньому або кваліфікаційному рівні відповідно до державних стандартів освіти.

Навчальний процес організовується з урахуванням можливостей сучасних інформаційних технологій навчання та орієнтується на формування освіченої, гармонійно розвиненої особистості, здатної до постійного оновлення наукових знань, професійної мобільності та швидкої адаптації до змін і розвитку в соціально-культурній сфері, в галузях техніки, технологій, системах управління та організації праці в умовах ринкової економіки.

Електронні освітні ресурси є складовою частиною навчального процесу. Вони мають навчально-методичне призначення та використовуються для забезпечення навчальної діяльності студентів і вважаються одним з головних елементів інформаційно-освітнього середовища. Одним з різновидів електронних освітніх ресурсів є освітні веб-ресурси.

Використання сучасних веб-технологій дає змогу суттєво вдосконалити

систему освіти, а отже, її подальша інформатизація — процес незворотний та обов'язковий. Освітні веб-ресурси у даній ситуації є оптимальним інструментом удосконалення професійної підготовки сучасних педагогів.

Розробка дистанційних курсів, забезпечення більшості вишівських спеціальностей дистанційними формами навчання вимагає грамотного вибору програмно-технічного оснащення цього процесу, знання існуючих платформ дистанційного навчання, вибору найбільш придатної для конкретного курсу. Паралельно має вестись робота із забезпечення дистанційного навчання навчальними засобами. Зараз серед електронних навчальних засобів, створених для дистанційного навчання, все більш значущими стають освітні веб-ресурси, зокрема веб-сайти, блоги.

Освітні веб-ресурси — це освітні електронні ресурси, що розміщенні у веб-просторі локальної чи глобальної мережі у вигляді різних форматів (текстового, графічного, архівного, аудіо- та відеоформатів) [1].

За функціональним призначенням вони поділяються на:

— навчальні;

— навчально-методичні;

— довідкові;

— нормативні;

— наукові;

— педагогічні;

— програмні засоби.

Таким чином, освітні веб-ресурси (сайти, блоги) передбачають інформаційно-презентаційну, консультативну, інформаційно-методичну, просвітницьку, навчальну підтримку діяльності суб’єктів взаємодії, відкривають нові можливості взаємодії з громадськістю та дозволяють:

— інтерактивно донести інформацію до аудиторії незалежно від її територіального місцезнаходження;

— оперативно висвітлювати діяльність веб-ресурсу на основі публікації новин, оглядів, каталогів видань, а також наукових, методичних і практичних матеріалів;

— використовувати сучасні засоби спілкування, такі як: електронна пошта,

інтерактивні конференції, форум — та ефективно організувати службу підтримки порталу;

— активізувати участь педагогічних працівників та учнів в Інтернет-олімпіадах, конкурсах, конференціях.

Застосування інформаційного простору в галузі освіти і безпосередньо в діяльності педагога стало загальною необхідністю. Освітні веб-ресурси можуть і повинні стати для педагогічної громадськості одним з пріоритетних засобів і способів самоосвіти; вони є ефективним організаційним елементом регіональної системи освіти, адже нові інформаційні технології впливають на всі компоненти освіти: зміст, методи та організаційні форми навчання, дозволяють вирішувати складні та актуальні завдання педагогу для забезпечення його інтелектуально-творчого розвитку.

Одним із шляхів підвищення якості освіти є підвищення ефективності контролю знань. Здійснення контролю в навчальному процесі має на меті виявити якість засвоєння знань, виміряти її величину та присвоїти цій якості певну оцінку.

Перевірка й оцінка знань студентів є активним процесом. Викладач не тільки пасивно реєструє фактичні знання студентів, а й впливає на хід і результати всього навчального процесу. Його завдання — знайти найефективніший засіб перевірки знань, щоб виявити досягнення студентів і стимулювати їх надалі оволодівати знаннями.

Необхідно створювати таку атмосферу, щоб студент обов'язково працював систематично, проявляючи наполегливість та вольові зусилля.

Одним із шляхів здійснення цієї мети є тестовий контроль знань.

Тестування або тестовий контроль знань — це засіб об’єктивного контролю ступеня досягнення кінцевих цілей підготовки студентів при якому рівень сформованості умінь встановлюється опосередковано за допомого ситуаційних тестів [2].

Переваги тестового контролю знань:

— об'єктивність — незалежність результатів тестування від особистих стосунків викладача та студента;

— простота процедури запису (введення) відповіді, незалежність оцінки від

техніки письма;

— кількісні критерії оцінки — наявність кількісних показників для визначення повноти та глибини засвоєння матеріалу;

— простота та формалізованість процедури визначення оцінки — можливість її здійснення людиною середньої кваліфікації або технічним пристроєм;

— чіткість та однозначність формулювання умов тестових завдань — що забезпечує однозначність сприйняття студентами їх змісту;

— рівні вимоги до знань та умінь студента шляхом використання в тесті завдань однакової складності, обсягу та змісту;

— забезпечення необхідної повноти охоплення знань та умінь, що контролюватимуться під час перевірки;

— можливість одночасної перевірки значної кількості студентів;

— можливість багаторазового повторення умов перевірки, для з'ясування змін в рівні підготовки;

— орієнтація на сучасні освітні технології — використання комп'ютерних навчальних і контролюючих систем;

— універсальність — охоплення всіх етапів процесу навчання;

— багатофункціональність — контроль, діагностика, корекція навчального процесу;

— охоплює контролем великий обсяг матеріалу;

— упродовж досить обмеженого часу може бути перевірена якість знань, навичок у зазначеної кількості студентів;

— можливий контроль знань, умінь, навичок на необхідному, заздалегідь запланованому рівні;

— реальним є самоконтроль;

— увага студента фіксується не на формуванні відповіді, а на осмисленні її суті;

— створюють умови для постійного зворотного зв'язку між студентом і викладачем.

Тестовий контроль знань як вирішення проблеми підвищення якості освіти приводить до висновків:

— урізноманітнення форм і засобів контролю якості знань студентів є актуальною проблемою, розв'язання якої викликане потребами педагогічної практики і перспективами інтеграції освітньої системи України з європейською спільнотою;

— використання тестових завдань, зокрема, з дисциплін, пов'язаних з комп'ютерною технікою та у комп'ютерній формі, практикується у більшості закордонних університетів і знаходить відгук в українській вищій школі [3];

— запровадження у навчальний процес вітчизняних навчально-методичних розробок здійснюється поступово завдяки співпраці викладачів вищих технічних і вищих педагогічних навчальних закладів;

— процес швидкого старіння наукової інформації потребує від викладачів постійного оновлення не лише лекційного матеріалу, але і відповідних контрольних тестових завдань і супровідних навчально-методичних матеріалів;

— студентів вищих навчальних закладів доцільно готувати до тестування систематично, на всіх курсах навчання та під час практики.

Традиційні методи перевірки й оцінки знань у поєднанні з новими технологіями відкривають перед викладачами широкі можливості. Оптимальним є контроль знань за допомогою тестування. Незаперечно, що підвищення якості навчання студентів безпосередньо пов'язане зі створенням і послідовним використанням системи тестового контролю засвоєння знань [4]. Позначена проблема набуває особливої актуальності в наш час, коли здійснюються суттєві кроки у напрямі інтеграції української системи вищої освіти до європейської університетської спільноти, набувають розповсюдження і підтримки ідеї так званого "Болонського процесу".

Якість освіти в першу чергу залежить від якості роботи викладача. Відповідно, питання контролю та оцінки якості роботи викладача є однією зі складних і важливих завдань в управлінні якістю освіти.

Однією з методик оцінки роботи викладачів може бути, наприклад, опитування студентів через певні проміжки часу або по закінченні вивчення окремих дисциплін [5].

Опитування — це метод збору соціологічної інформації про досліджуваний об'єкт під час безпосереднього або опосередкованого спілкування того хто опитує з

особою, яка відповідає на питання.

Використання опитування студентів для визначенні якості викладання дозволить більш ефективно вирішувати питання контролю та оцінки якості роботи викладача, воно є умовою вдосконалення професійних знань і педагогічної майстерності викладача [6]. Для студентів опитування дає можливість не просто оцінити викладача, а й оцінити себе як учасника освітнього процесу. Також опитування можуть допомогти студентам і викладачам у організаційних питаннях.

Використання постійного контролю знань у навчальному процесі значно підвищує мотивацію навчання, а опитування покращують якість освітнього процесу, однак для проведення тестування чи опитування викладачі мають виконати велику кількість роботи для формування завдань та організації процесу. Тому важливим рішенням цієї проблеми є використання веб-ресурсів для організації тестів і опитувань, що дозволяє дуже швидко здійснювати контроль, максимально автоматизувати процеси тестування і опитування, а також використовувати тести і опитування у дистанційному навчанні.

3. МЕТОДИ ТА ЗАСОБИ РЕАЛІЗАЦІЇ WEB-РЕСУРСУ

Аналізуючи поставлену задачу та методи її вирішення, було вирішено розроблювати програмний комплекс на основі веб-технологій. Головною перевагою веб-застосунку перед іншими варіантами є його універсальність і можливість використання на будь-яких пристроях без портування на цільову операційну систему (браузер і його віртуальна машина виступає як цільова універсальна операційна система і комп'ютер).

3.1. Вибір архітектури програмного комплексу

Для реалізації поставленої задачі було вирішено використовувати триланкову архітектуру, яка складається з таких компонентів: клієнт, сервер і база даних. Схема даної архітектури зображена на рисунку 3.1.

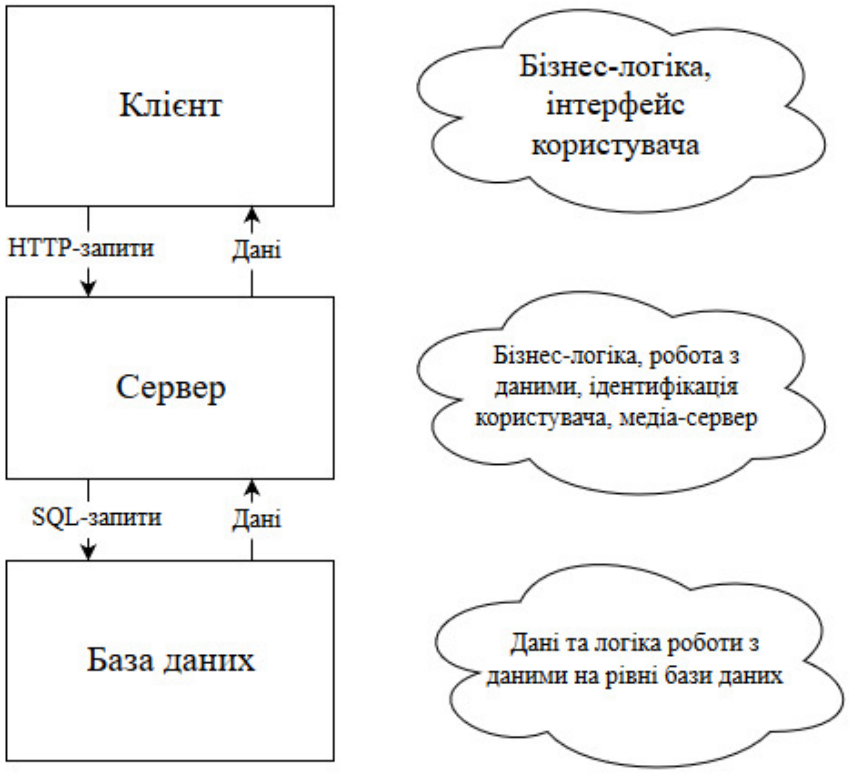

Рисунок 3.1 — Триланкова архітектура програмного комплексу

Головним центром програмного комплексу є сервер. У ньому зосереджена основна бізнес-логіка та логіка доступу до бази даних. За допомогою серверу відбувається ідентифікація користувача для надання індивідуального доступу до програмного застосунку. Сервер є єдиним зв’язком між користувачем та базою даних, щоб унеможливити пошкодження даних та їх використання не за призначенням. Для того, щоб користуватися програмою, потрібно бути авторизованим у системі, тому дана логіка реалізуються на рівні серверу, тому що на рівні користувача можлива підміна прав доступу та інші методи неконтрольованого доступу до даних.

Під час користування програмою, користувач взаємодіє з клієнтським додатком, яким є веб-сайт в даному випадку. На рівні користувача реалізований інтерфейс, за допомогою якого відбувається налаштування програми та перегляд результатів роботи. Також на користувацькому рівні відбувається попередня обробка даних перед відправленням на сервер і також опрацювання результатів від сервера. Ще на цьому рівні відбувається перший етап аутентифікації користувача для обмеження неконтрольованого доступу до програми.

Важливою задачею рівня бази даних є забезпечення збереження даних, які сервер зберігає для подальшого використання. Також забезпечується цілісність даних за допомогою зовнішніх зв’язків та ключів. На рівні бази даних також можна реалізовувати деяку бізнес-логіку, яка не потребує використання зовнішніх джерел даних окрім самої бази даних та її таблиць.

3.2. Опис архітектури серверу

Для реалізації серверу було використано API, який є інтерфейсом прикладного програмування.

Прикладний програмний інтерфейс API — це набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено — це набір чітко визначених методів для взаємодії різних компонентів. Програмний інтерфейс API надає розробнику засоби для швидкої розробки програмного забезпечення. Програмний інтерфейс API може бути для веб-базованих систем,

операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. Програмісти використовують переваги API у функціональності, таким чином їм не доводиться розробляти все з нуля. Програмний інтерфейс API є абстрактним поняттям — програмне забезпечення, що пропонує деякий API, часто називають реалізацією даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, API визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON) [7]. У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів орієнтованої архітектури (ROA). Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у веб, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

3.3. Опис архітектури клієнтського застосунку

Для реалізації клієнтського застосунку було використано фреймворк, в основу якого було покладено шаблон проектування MVVM (рисунок 3.2).

Шаблон MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), відомої як модель (можна також сказати, що це відокремлення представлення від моделі) [8]. Модель представлення є частиною,

яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати патерн медіатор, організовуючи доступ до бек-енд логіки навколо множини правил використання, які підтримуються представленням.

Шаблон MVVM зручно використовувати у тих випадках, коли на платформі, де ведеться розробка, присутнє “зв'язування даних”.

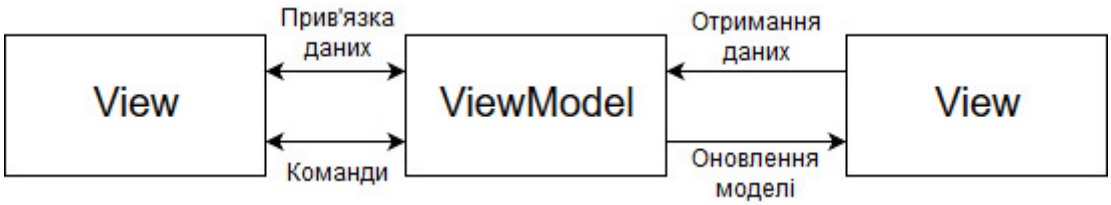

Рисунок 3.2 — Схема роботи шаблону MVVM

Шаблон MVVM складається з таких частин:

— Модель (Model), як і в класичному шаблоні MVC, Модель являє собою фундаментальні дані, що необхідні для роботи застосунку;

— Представлення (View) як і в класичному шаблоні MVC, Вигляд — це графічний інтерфейс, тобто вікно, кнопки тощо;

— Модель вигляду (ViewModel, що означає “Model of View”) з одного боку є абстракцією Вигляду, а з іншого надає обгортку даних з Моделі, які мають зв'язуватись.

3.4. Опис інструментів розробки

Програмний комплекс побудований за принципами триланкової архітектури побудови програм. Кожен рівень цієї архітектури реалізовано з використанням різних технологій і головною ціллю було створення мультиплатформного рішення з використанням відкритих технологій.

Для клієнтського рівня було використано такий набір технологій: мова програмування JavaScript, фреймворк для побудови веб-застосунків Vue.js, фреймворк для створення графічних інтерфейсів Vuetify.

Для серверного рівня було використано такі технології: мова програмування PHP, серверна платформа Open Server, веб-сервер Apache.

Для рівня бази даних було обрано MySQL.

Також під час створення веб-системи була використана технологія AJAX.

3.4.1. Мова програмування JavaScript, фреймворки Vue.js та Vuetify

Мова програмування JavaScript є динамічною, об'єктно-орієнтованою прототипною мовою програмування. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

Мову JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

— написання сценаріїв веб-сторінок для надання їм інтерактивності;

— створення односторінкових веб-застосунків (React, AngularJS, Vue.js);

— програмування на стороні сервера (Node.js);

— стаціонарних застосунків (Electron, NW.js);

— мобільних застосунків (React Native, Cordova);

— сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);

— всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme [9].

При використанні в рамках технології DHTML JavaScript-код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги <script></script> хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript.

При розробці великих і нетривіальних веб-застосунків з використанням JavaScript, критично важливим є доступ до інструментів відлагодження. Оскільки браузери, від різних виробників, дещо відрізняються у поведінці JavaScript і реалізації Об'єктної моделі документа, необхідно мати відлагоджувач для кожного браузера, якщо веб-застосування орієнтовано на нього.

Оскільки JavaScript є інтерпретатором, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути дуже уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій.

Кожен блок сценарію інтерпретатор розбирає окремо. На веб-сторінках, коли треба комбінувати блоки JavaScript та HTML, синтаксичні помилки знайти легше, якщо тримати функції сценарію в окремому блоці коду, або (ще краще) використовувати багато малих пов'язаних .js файлів. В такий спосіб синтаксична помилка не спричинятиме «падіння» цілої сторінки, і можна надати допомогу, елегантно вийшовши зі сторінки.

Для JavaScript існують кілька фреймворків, серед яких є Vue.js та Vuetify.

Фреймворк Vue.js — JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних. Фреймворк Vue використовує синтаксис шаблонів на основі HTML, що дозволяє декларативно зв'язувати рендеринг DOM з основними екземплярами даних в Vue [10]. Одна із найвиразніших особливостей Vue — це ненав’язлива реактивна система. Моделі це прості JavaScript об’єкти. Це робить

керування станами дуже простим та інтуїтивним [11].

Фреймворк Vuetify — це набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків [12].

3.4.2. Мова програмування PHP

Мова програмування PHP є скриптовою мовою програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. Мова PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). Мова PHP підтримується переважною більшістю хостинг-провайдерів. Мова PHP — проект відкритого програмного забезпечення.

Мова PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок [13].

У PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.

Завдяки стандарту відкритого інтерфейсу зв'язку з базами даних можна підключатися до всіх баз даних, до яких існує драйвер.

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів для розробки веб-додатків [14]. Основні з них:

— автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;

— взаємодія з великою кількістю різних систем управління базами даних (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes , DB + +, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Інтерфейс PDO);

— автоматизована відправка HTTP-заголовків;

— робота з HTTP-авторизацією;

— робота з cookies і сесіями;

— робота з локальними і віддаленими файлами, сокетами;

— обробка файлів, що завантажуються на сервер;

— робота з XForms.

Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустріне парну екрануючу послідовність (?>).

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах мовами C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. Мова програмування PHP є мовою, що поєднує переваги Perl та C і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом [15]. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед веб-програмістів, що в наш час є найпопулярнішою мовою для створення веб-застосунків (скриптів).

Велика різноманітність функцій PHP дають можливість уникнути написання багаторядкових призначених для користувача функцій на C або Pascal. Мова PHP — мова з універсальним і зрозумілим синтаксисом та спеціально спрямована на роботу в Інтернеті. Також, важливо зазначити наявність вихідного коду та безкоштовність. Важливою перевагою PHP є те, що ця мова належить до інтерпретуючих. Це дозволяє обробляти сценарії з достатньо високою швидкістю [16].

Одним із застосувань мови PHP є розробка парсеру — синтаксичного аналізатору, програми або частини програми, що виконує синтаксичний аналіз. За заданим списком URL-посилань, програма переходить на вказаний сайт та завантажує HTML-сторінку, після чого проводить її розбір та опрацьовує дані згідно з поставленою задачею [17].

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і Web. Важливою перевагою

PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте хоч би що робили розробники PHP, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-застосунків.

3.4.3. Серверна платформа Open Server та веб-сервер Apache

Платформа Open Server — це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників із врахуванням їх рекомендацій.

Програмний комплекс має великий набір серверного програмного забезпечення, зручний, багатофункціональний інтерфейс, володіє можливостями адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а також для надання веб-сервісів у локальних мережах.

Платформа Open Server включає у себе Apache, nginx, PHP, MySQL, phpMyAdmin, FTP-сервер FileZilla.

Веб-сервер Apache є відкритим веб-сервером Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Продукт підтримує безліч можливостей, багато з яких реалізовані як скомпільовані модулі, які розширюють основні функціональні можливості. Вони різняться від серверної підтримки мов програмування до схем аутентифікації. Існують інтерфейси для підтримки мов програмування Perl, Python, Tcl і PHP. Функції віртуального хостингу дозволяють одній інсталяції Apache обслуговувати різні веб-сайти.

Веб-сервер Apache передусім використовується для передачі через HTTP статичних та динамічних веб-сторінок у всесвітній павутині. Продукт може працювати як кешувальний проксі-сервер, що дозволяє істотно підвищити продуктивність роботи користувачів локальної мережі при роботі з документами,

розташованими в Інтернет. Можна задавати такі параметри і налаштування проксі-сервера:

— типи файлів, які необхідно кешувати або навпаки, не включати в кеш;

— максимальний обсяг дискового простору, відведений під кеш;

— періодичний перегляд і індексування бази даних кеша з метою вивільнення дискового простору шляхом видалення застарілих об'єктів.

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation, без участі сторонніх програмістів.

Теоретично, ядро Apache може функціонувати в чистому вигляді, без використання модулів. Однак, функціональність такого рішення вкрай обмежена.

Ядро Apache повністю написано на мові програмування C.

Система конфігурації Apache заснована на текстових конфігураційних файлах. Має три умовних рівня конфігурації:

1. конфігурація сервера (httpd.conf);
2. конфігурація віртуального хоста (httpd.conf з версії 2.2 extra/httpd-vhosts.conf);
3. конфігурація рівня директорії (.htaccess).

Веб-сервер має власну мову конфігураційних файлів, заснований на блоках директив. Практично всі параметри ядра можуть бути змінені через конфігураційні файли. Більша частина модулів має власні параметри [18].

Частина модулів використовує в своїй роботі конфігураційні файли операційної системи (наприклад /etc/passwd і /etc/hosts). Крім цього, параметри можуть бути задані через ключі командного рядка [19].

Для веб-сервера Apache існує безліч моделей симетричного мультипроцесування:

— worker — гібридна мультипроцесорна-багатонитева модель. Зберігаючи стабільність мультипроцесорних рішень, вона дозволяє обслуговувати велику кількість клієнтів з мінімальним використанням ресурсів;

— pre-fork — модель симетричного мультипроцесування, заснована на попередньому створенні окремих процесів;

— perchild — гібридна модель, з фіксованою кількістю процесів;

— netware — багатонитева модель, оптимізована для роботи в середовищі NetWare;

— winnt — багатонитева модель, створена для операційної системи Microsoft Windows;

— Apache-ITK — модель симетричного мультипроцесування, заснована на моделі prefork. Дозволяє запуск кожного віртуального хоста під окремими UID та GID;

— peruser — модель, створена на базі MPM perchild. Дозволяє запуск кожного віртуального хоста під окремими UID та GID. Не використовує нитi.

3.4.4. Система керування базами даних MySQL

Система MySQL — вільна система керування реляційними базами даних, а також компактний багатопотоковий сервер баз даних, який характеризується високою швидкістю, стійкістю і простотою використання. У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається виграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. Мову SQL як частину системи MySQL можна охарактеризувати як мову структурованих запитів, що використовується для доступу до баз даних.

Система MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому [20].

Можливості сервера MySQL:

— простота у встановленні та використанні;

— підтримується необмежена кількість користувачів, що одночасно працюють із БД;

— кількість рядків у таблицях може досягати 50 млн;

— висока швидкість виконання команд;

— наявність простої і ефективної системи безпеки.

Система MySQL — це ПЗ з відкритим кодом. Застосовувати його і модифікувати може будь-хто. При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

Система MySQL складається з двох частин: серверної і клієнтської.

Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту) [21].

Структура MySQL трирівнева: бази даних — таблиці — записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно таблиця являє собою сукупність записів. А запис — це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці — в межах бази даних, поля — в межах таблиці. Один сервер MySQL може підтримувати одразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем.

База даних з точки зору MySQL — це звичайний каталог, що містить файли певного формату — таблиці. Таблиці складаються із записів, а записи, у свою чергу, складаються з полів. Поле має два атрибути — ім'я і тип.

Основною функцією для з'єднання з сервером MySQL є mysql_connect (), яка підключає скрипт до сервера баз даних MySQL та виконується авторизацію користувача базою даних [22]. Синтаксис у даної функції такий: mysql_connect ([string $hostname] [, string $user] [, string $password]);

Для закриття з'єднання призначена функція mysql_close (int $ connection_id). Взагалі, підключення можна і не закривати — воно буде закрито автоматично по завершенні роботи PHP скрипта. Однак, це поганий стиль. Якщо кількість з'єднань більше одного, вказується ідентифікатор $ connection_id того з'єднання, яке необхідно

закрити.

Функція mysql_connect () встановлює звичайне з'єднання з MySQL. Однак, PHP підтримує постійні з'єднання — для цього використовують функцію mysql_pconnect (). Аргументи цієї функції такі ж, як і у mysql_connect (). Різниця між постійним і простим з'єднанням в тому, що постійне з'єднання не закривається після завершення роботи скрипта, навіть якщо скрипт викликав функцію mysql_close ().

Всі запити до поточної базі даних відправляються функцією mysql_query (). Цій функції потрібно передати один параметр — текст запиту. Текст запиту модет містити пробільні символи і символи нового рядка (\n). Текст повинен бути складений за правилами синтаксису SQL [23].

Якщо запит, виконаний за допомогою функції mysql_query () успішно виконався, то в результаті клієнт отримає набір записів, який може бути оброблений функціями PHP. Для обробки великих наборів записів рекомендується використовувати функції mysql_fecth_row (), mysql_fecth_array (), і.т.д.

Функція mysql_fecth_row (int $ res) повертає масив, що містить дані обробленого рядка, або FALSE, якщо рядків більше немає. Вона обробляє один ряд результату, на який посилається переданий покажчик. Ряд повертається в масиві. Кожна колонка розташована в наступній комірці масиву. Масив починається з індексу 0. Наступні виклики функції mysql_fetch_row () повернуть наступні рядки або FALSE, якщо рядків не залишилося. Імена полів, що повертаються цією функцією, чутливі до регістру.

Функція mysql_fecth_array (int $ res [, int $ result_type]) повертає асоціативний масив, заданий необов'язковим параметром $ result_type, який може приймати наступні значення:

— MYSQL_ASSOC — повертає асоціативний масив;

— MYSQL_NUM — повертає масив з числовими індексами, як у функції mysql_fecth_row ();

— MYSQL_BOTH — повертає масив з подвійними індексами, тобто можна працювати з ним, як з асоціативним масивом і як зі списком (MYSQL_BOTH — це значення за замовчуванням для параметра $ result_type).

3.4.5. Технологія AJAX

Технологія AJAX — підхід до побудови інтерактивних користувацьких інтерфейсів веб-додатків, що полягає в «фоновому» обміні даними браузера з веб-сервером. В результаті, при оновленні даних, веб-сторінка не перезавантажується повністю, і веб-додатки стають швидшими і зручнішими.

Технологія AJAX — не самостійна технологія, а концепція використання декількох суміжних технологій. Вона базується на двох основних принципах:

1. використання технології динамічного звернення до серверу «на льоту», без перезавантаження всієї сторінки повністю, наприклад:

— з використанням XMLHttpRequest (основний об’єкт);

— через динамічне створення дочірніх фреймів;

— через динамічне створіння тега <script>;

2. використання DHTML для динамічної зміни змісту сторінки.

В якості формату передачі даних зазвичай використовуються JSON або XML.

Переваги:

— економія трафіку (використання AJAX дозволяє значно скоротити трафік при роботі з веб-додатком завдяки тому, що часто замість завантаження всієї сторінки достатньо завантажити тільки змінилася частина, як правило, досить невелику;

— зменшення навантаження на сервер (AJAX дозволяє дещо понизити навантаження на сервер);

— прискорення реакції інтерфейсу (оскільки потрібно завантажити тільки змінилася частина, користувач бачить результат своїх дій швидше).

Недоліки:

— відсутність інтеграції зі стандартними інструментами браузера (динамічно створювані сторінки не реєструються браузером в історії відвідин сторінок, тому не працює кнопка «Назад», що надає користувачам можливість повернутися до переглянутих раніше сторінок, але існують скрипти, які можуть вирішити цю проблему, інший недолік зміни вмісту сторінки при постійному URL полягає в неможливості збереження закладки на бажаний матеріал. Частково вирішити ці проблеми можна за допомогою динамічної зміни ідентифікатора фрагмента (частини

URL після #), що дозволяють багато браузери);

— динамічно завантажувати вміст недоступно пошуковикам (якщо не перевіряти запит, звичайний він або XMLHttpRequest), пошукові машини не можуть виконувати JavaScript, тому розробники повинні поклопотатися про альтернативні способи доступу до вмісту сайту;

— старі методи обліку статистики сайтів стають неактуальними (багато сервісів статистики ведуть облік переглядів нових сторінок сайту. Для сайтів, сторінки яких широко використовують AJAX, така статистика втрачає актуальність);

— ускладнення проекту (перерозподіляється логіка обробки даних — відбувається виділення і часткове перенесення на сторону клієнта процесів первинного форматування даних. Це ускладнює контроль цілісності форматів і типів. Кінцевий ефект технології може бути знівельовано не обгрунтованим зростанням витрат на кодування і управління проектом, а також ризиком зниження доступності сервісу для кінцевих користувачів);

— потрібен включений JavaScript в браузері.

Невід'ємною складовою технології AJAX є XMLHttpRequest — API запит веб-клієнта (браузера) до веб-сервера за протоколом HTTP у фоновому режимі, для мов програмування JavaScript, JScript, VBScript і подібних. Використовується для синхронного або асинхронного обміну інформацією в довільному текстовому форматі, зокрема в форматах XML або JSON [7].

3.5. Обґрунтування вибору програмної реалізації

При проектуванні системи було вивчено та проаналізовано предметну область та вимоги замовника. Після ретельного аналізу було вирішено розроблювати програмний продукт, який заснований на веб-технологіях для використання за допомогою веб-браузера.

Технології, які використовуються на сервері, були обрані за принципом зручності у використанні, актуальності в наш час та можливістю виконання на будь-якій операційній системі. Платформа Open Server є дуже зручною серверною

платформою, яка дозволяє дуже швидко налагодити доступ до веб-серверу. Сервер Apache є самим веб-сервером, який дозволяє обслуговувати різні сайти за допомогою веб-хостингу. Для доступу до бази даних було використано мову програмування PHP.

На клієнтському рівні було обрано технології, які задовольняють такі ж умови, як і серверні, але з поправкою на виконання в браузері. Фреймворк Vue.js було обрано через те, що за допомогою нього можливо створювати складні графічні інтерфейси, які легко модифікувати, тестувати та розширювати в подальших циклах розробки програмного забезпечення. Мова JavaScript була обрана через те, що вона є прототипною скриптовою мовою програмування з динамічною типізацією. Для того щоб стилізувати клієнтський додаток було вирішено використати фреймворк Vuetify через його багату бібліотеку готових інтерфейсів та вбудовану підтримку та масштабування до розмірів мобільного телефону.

Базою даних було обрано MySQL через те, що це проект з відкритим вихідним кодом і надзвичайно великою підтримкою з боку розробників. Також важливою властивістю є велика швидкість роботи, надійність та кількість вбудованих можливостей.

Також для створення системи була обрана технологія AJAX через її можливість використовувати API-запит веб-клієнта (браузера) до веб-сервера за протоколом HTTP у фоновому режимі.

Дані технології в сукупності дають змогу збудувати якісний та надійний продукт, який захищений від патентних позовів з боку розробників, бо всі ці технології покриті ліцензіями, які виключають таку можливість і надають доступ до вихідних кодів даних проектів.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1. Архітектурна схема програмної системи

Система організації та проведення тестів і опитувань у навчальному процесі буде складатися з кількох модулів, кожен з яких буде розбиватися на різну кількість функціональних підблоків. Головним модулем буде кабінет користувача, який одночасно буде елементом компонування системи. Така структура, дозволить легко та інтуїтивно користуватися системою за рахунок того, що модулі розташовуються у порядку, який зазвичай використовують для вирішення схожих задач.

На рисунку 4.1 наведена схема структури системи, на якій розташовані програмні модулі.

Рисунок 4.1 — Схема структури системи

У систему заносяться вхідні дані — відповіді на питання, які обробляються, після чого система виконує підрахунки і виводить результат — оцінка за тест або загальні результати опитування.

4.2. Опис функціональності системи

Програмний застосунок для організації та проведення тестів і опитувань у навчальному процесі містить у собі трьох головних акторів — користувачів системи;

На рисунку 4.2 представлена діаграма прецедентів, яка описує функції та дії акторів у системі.

Рисунок 4.2 — Діаграма прецедентів системи

За діаграмою існують три типи користувачів: студент, викладач та адміністратор. Викладач створює тести і опитування, студент має їх проходити, а адміністратор реєструє викладачів та студентів у систему, а також може видалити застарілу інформацію. Всі три типи користувачів можуть переглядати результати тестів і опитувань.

4.3. Опис бази даних

Весь зміст системи а також її основні дані, тобто дані про тести і опитування зберігається в базі даних. Концептуальна модель бази даних має наступний вигляд (рисунок 4.3):

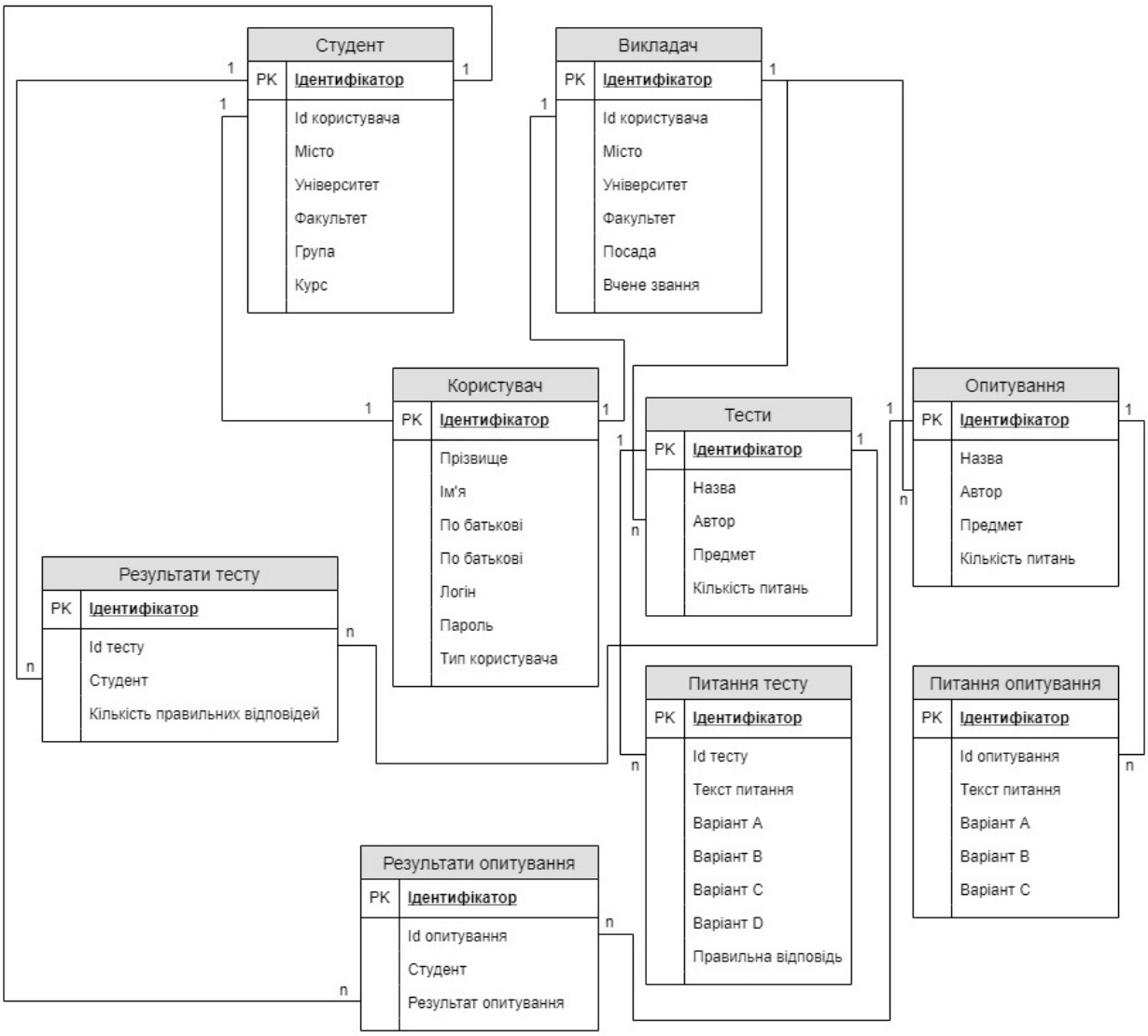

Рисунок 4.3 — Концептуальна модель бази даних

База даних складається з наступних таблиць: “Користувач”, “Студент”, “Викладач”, “Тести”, “Опитування”, “Питання тесту” , “Питання опитування”, “Результати тесту”, “Результати опитування”.

Структура таблиці “Користувач” представлена наступним чином (таблиця 4.1):

Таблиця 4.1 — Структура таблиці “Користувач”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
Surname	varchar	Прізвище
Name	varchar	Ім’я
Patronymic	varchar	По батькові
UserName	varchar	Логін користувача
Password	varchar	Пароль
Status	int	Тип користувача

Таблиця складається з таких полів як Ідентифікатор, Прізвище, Ім’я, По батькові, Логін, Пароль та Тип користувача.

Таблиця “Студент” містить зв’язок з таблицею “Користувач” по ідентифікатору користувача, так як виражає детальні дані, що стосуються кожного студента (таблиця 4.2).

Таблиця 4.2 — Структура таблиці “Студент”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
IdUser	int	Id користувача
IdTown	int	Місто
IdUniversity	int	Університет
IdFaculty	int	Факультет
IdGroup	int	Група
Course	int	Курс

Таблиця “Викладач” містить зв’язок з таблицею “Користувач” по ідентифікатору користувача, так як виражає детальні дані, що стосуються кожного викладача (таблиця 4.3).

Таблиця 4.3 — Структура таблиці “Викладач”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
IdUser	int	Id користувача
IdTown	int	Місто
IdUniversity	int	Університет
IdFaculty	int	Факультет
Position	int	Посада
AcademicTitle	int	Вчене звання

Таблиця “Тести” містить такі поля як Ідентифікатор, Назва, Автор, Предмет та Кількість питань. Дана таблиця має зв’язок з таблицею “Викладач” по ідентифікатору автора, так як виражає детальні дані, що стосуються тесту, створеного викладачем (таблиця 4.4).

Таблиця 4.4 — Структура таблиці “Тести”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
Name	varchar	Назва
IdAuthor	int	Автор
IdSubject	int	Предмет
AmountQuestions	int	Кількість питань

Таблиця “Опитування” містить такі поля як Ідентифікатор, Назва, Автор, Предмет та Кількість питань. Дана таблиця має зв’язок з таблицею “Викладач” по ідентифікатору автора, так як виражає детальні дані, що стосуються опитування, створеного викладачем (таблиця 4.5).

Таблиця 4.5 — Структура таблиці “Опитування”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
Name	varchar	Назва
IdAuthor	int	Автор
IdSubject	int	Предмет
AmountQuestions	int	Кількість питань

Таблиця “Питання тесту” містить такі поля як Ідентифікатор поля, Ідентифікатор тесту, Текст питання, Правильна відповідь та значення в кожному з чотирьох варіантів відповідей. Дана таблиця має зв’язок з таблицею “Тест” по ідентифікатору тесту (таблиця 4.6).

Таблиця 4.6 — Структура таблиці “Питання тесту”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
IdTest	int	Ідентифікатор тесту
TextQuestion	varchar	Текст питання
VariantA	varchar	Варіант А
VariantB	varchar	Варіант В
VariantC	varchar	Варіант С
VariantD	varchar	Варіант D
RightAnswer	int	Правильна відповідь

Таблиця “Питання опитування” містить такі поля як Ідентифікатор поля, Ідентифікатор опитування, Текст питання, Правильна відповідь та значення в кожному з трьох варіантів відповідей. Дана таблиця має зв’язок з таблицею “Опитування” по ідентифікатору опитування (таблиця 4.7).

Таблиця 4.7 — Структура таблиці “Питання опитування”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
IdPoll	int	Ідентифікатор тесту
TextQuestion	varchar	Текст питання
VariantA	varchar	Варіант А
VariantB	varchar	Варіант В
VariantC	varchar	Варіант С

Таблиця “Результати тесту” містить такі поля як Ідентифікатор поля, Ідентифікатор тесту, Студент та Кількість правильних відповідей. Дана таблиця має зв’язок з таблицею “Тести” по ідентифікатору тесту.

Також таблиця має зв’язок з таблицею “Студент” по ідентифікатору студента (таблиця 4.8).

Таблиця 4.8 — Структура таблиці “Результати тесту”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
IdTest	int	Ідентифікатор тесту
IdStudent	int	Студент
AmountRightAnswers	int	Кількість правильних відповідей

Таблиця “Результати опитування” містить такі поля як Ідентифікатор поля, Ідентифікатор опитування, Студент та Результат опитування.

Дана таблиця має зв’язок з таблицею “Опитування” по ідентифікатору опитування.

Також таблиця має зв’язок з таблицею “Студент” по ідентифікатору студента (таблиця 4.9).

Таблиця 4.9 — Структура таблиці “Результати опитування”

Ім’я поля	**Тип і розмір поля**	**Опис поля**
Id	int	Первинний ключ
IdPoll	int	Ідентифікатор опитування
IdStudent	int	Студент
Result	int	Результат опитування

Також у базі даних є таблиці міст, університетів, факультетів, предметів, навчальних груп, посад викладачів та вчених звань, які складаються з двох полів: Ідентифікатор та Назва. З ними мають зв’язок таблиці “Студент” і “Викладач”.

4.4. Розробка кабінету користувача

Кабінет користувача — це основне робоче місце користувача в системі. Він включає в себе інші підмодулі, які виконують основні функції системи, та є елементом компонування в системі.

Підмодулі кабінету користувача:

До цих модулів належать наступні:

— модуль керування тестами;

— модуль керування опитуваннями;

— модуль керування даними викладачів та студентів.

4.4.1. Модуль керування тестами

Даний модуль дозволяє користувачу додатку керувати наборами тестів, а саме:

— створювати нові тести (для викладачів);

— проходити тестування (для студентів);

— переглядати результати тестів;

— видаляти застарілі дані про тести (для адміністраторів).

4.4.2. Модуль керування опитуваннями

Даний модуль дозволяє користувачу додатку керувати наборами опитувань, а саме:

— створювати нові опитування (для викладачів);

— проходити опитування (для студентів);

— переглядати результати опитувань;

— видаляти застарілі дані про опитування (для адміністраторів).

4.4.3. Модуль керування даними викладачів та студентів

Даний модуль дозволяє користувачу додатку керувати даними викладачів та студентів, а саме:

— реєструвати у системі нових викладачів та студентів (для адміністраторів);

— змінювати логін і пароль до облікового запису;

— змінювати основні дані викладача або студента (для адміністраторів);

— видаляти облікові записи викладачів та студентів (для адміністраторів).

5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА

Програмний комплекс розроблений з використанням веб-технологій і тому працює в браузерах, які підтримують актуальні веб-стандарти.

5.1. Інсталяція та системні вимоги

Оскільки застосунок працює з використанням веб-технологій, він не потребує встановлення на пристрій користувача. Проте, для використання необхідний веб-браузер, який підтримує актуальні веб-стандарти. Також необхідно мати стабільний доступ до інтернету з швидкістю не менше ніж 50 мбіт за секунду. Для використання можливостей програми з проведення тестів і опитувань необхідно дати клієнтському додатку відповідний дозвіл.

5.2. Інструкція з використання програмного продукту

При вході на клієнтський додаток, користувачеві необхідно авторизуватися в системі щоб почати працювати в системі. На рисунку 5.1 зображена форма авторизації.

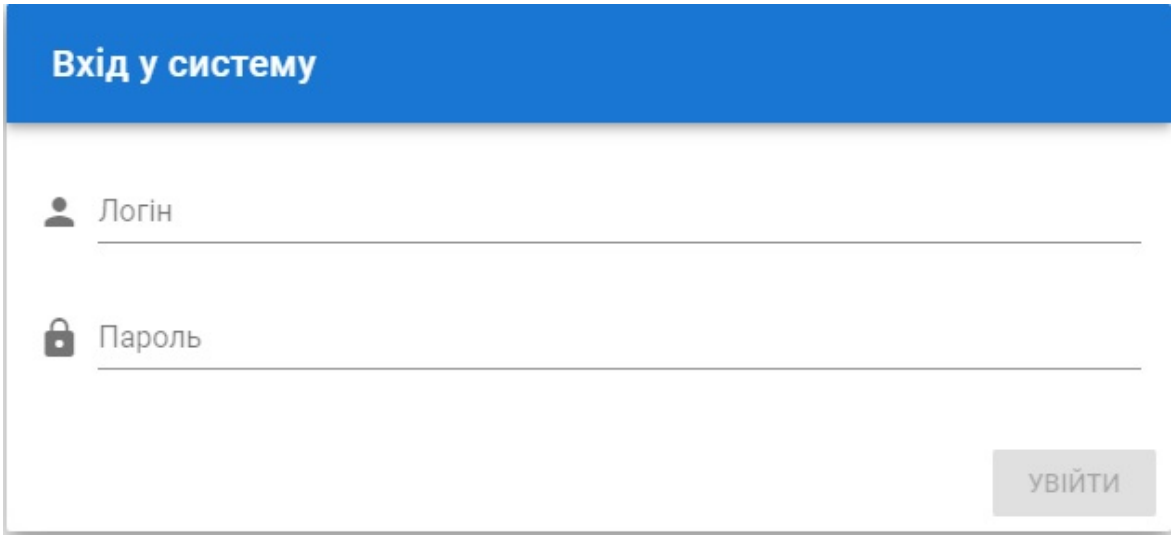

Рисунок 5.1 — Форма авторизації

У випадку якщо користувач ще ніколи не користувався системою або хоче створити новий профіль, то йому необхідно зареєструватися в системі. На рисунку 5.2 зображена форма реєстрації.

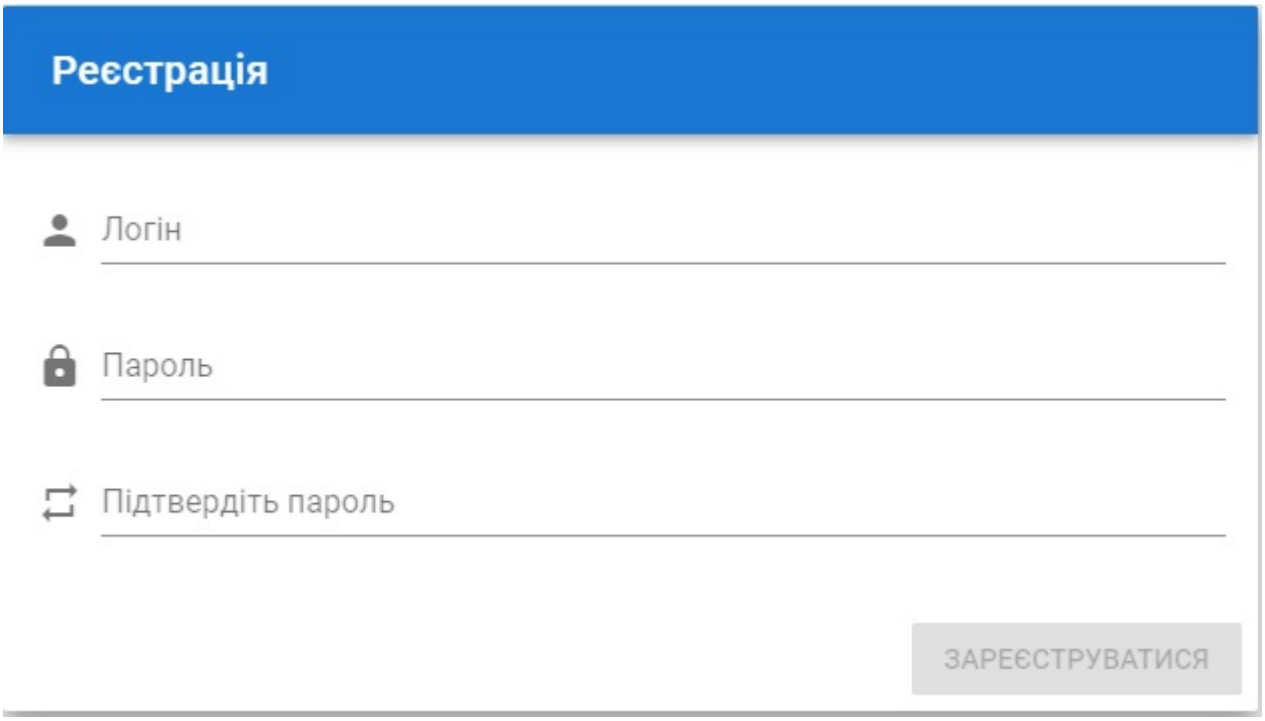

Рисунок 5.2 — Форма реєстрації

Після того, як користувач авторизувався в системі, він отримує доступ до головного меню системи. За допомогою цього меню користувач має доступ до усіх сторінок системи, або має можливість вийти з свого профілю. Також він може перейти до свого основного робочого простору — кабінету користувача. На рисунку 5.3 зображене головне меню системи.

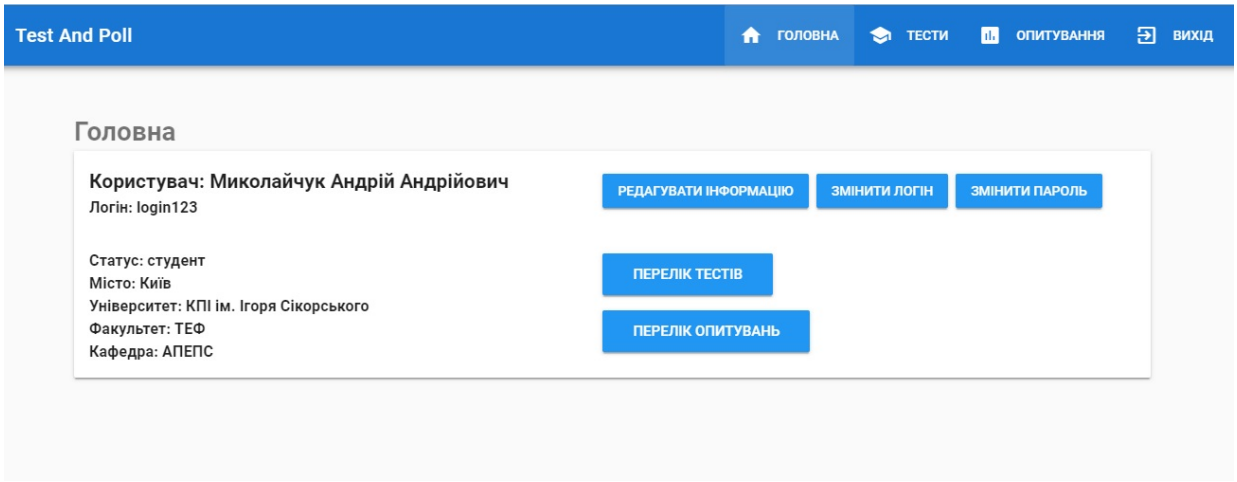

Рисунок 5.3 — Головне меню системи

Розроблено також пункти роботи з системою: “Тести”, “Опитування”, “Користувачі”.

За допомогою пункту меню “Тести” можна переглянути результати пройдених тестів, а також створити новий тест або пройти його (в залежності від типу користувача).

За допомогою пункту меню “Опитування” можна переглянути результати пройдених опитувань, а також створити нове опитування або пройти його (в залежності від типу користувача).

За допомогою пункту меню “Користувачі” адміністратор може реєструвати нових викладачів і студентів, переглянути список зареєстрованих викладачів і студентів, змінювати їх дані або видаляти їх облікові записи.

5.2.1. Пункт меню “Тести”

Для перегляду списку пройдених тестів необхідно натиснути на кнопку “Тести” (рисунок 5.4).

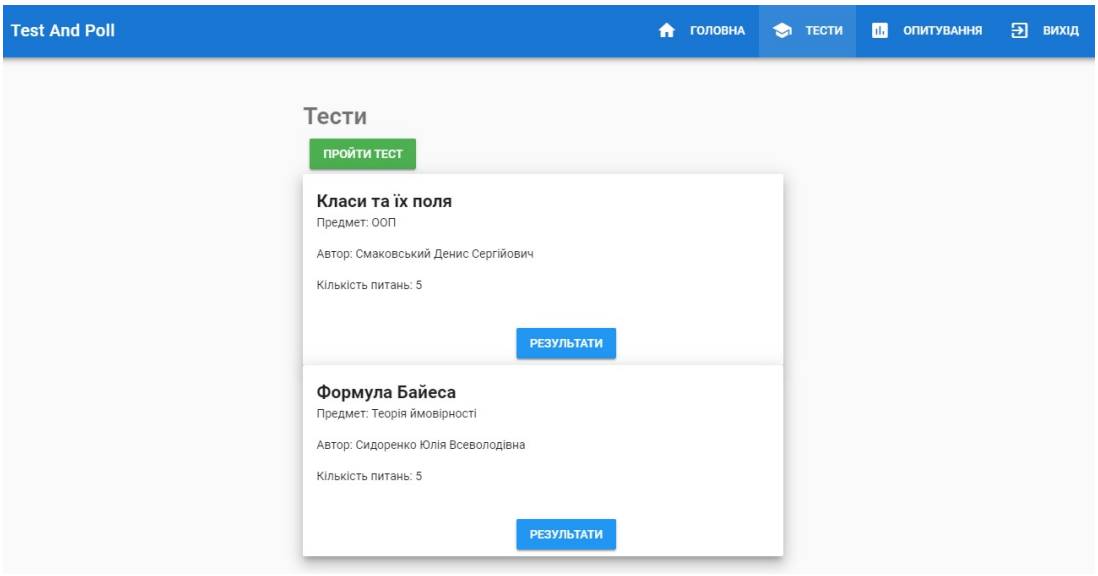

Рисунок 5.4 — Пункт меню “Тести”

Щоб переглянути результат пройденого тесту, потрібно на елементі списку натиснути кнопку “Результати”.

Для створення нового тесту викладачеві достатньо натиснути кнопку “Створити тест”, яка доступна тільки йому.

Для проходження нового тесту студенту необхідно натиснути на кнопку “Пройти тест”, яка доступна лише для нього.

5.2.2. Пункт меню “Опитування”

Для перегляду списку пройдених опитувань необхідно натиснути на кнопку “Опитування” (рисунок 5.5).

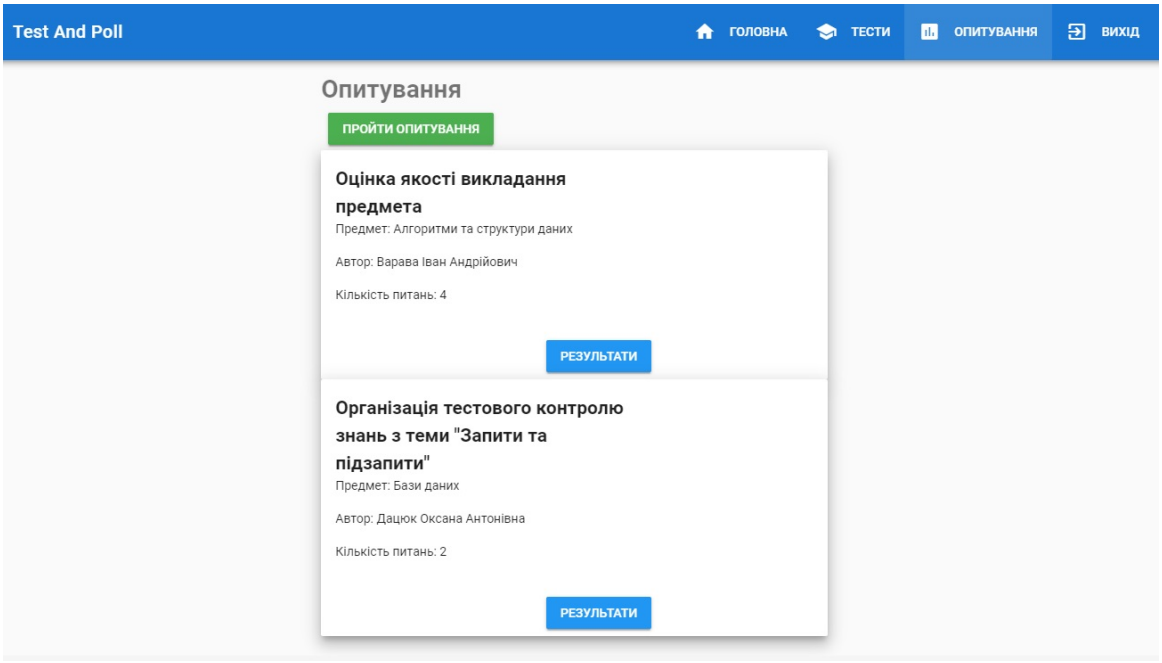

Рисунок 5.5 — Пункт меню “Опитування”

Щоб переглянути результат пройденого опитування, потрібно на елементі списку натиснути кнопку “Результати”.

Для створення нового опитування викладачеві достатньо натиснути кнопку “Створити опитування”, яка доступна тільки йому.

Для проходження нового опитування студенту необхідно натиснути на кнопку “Пройти опитування”, яка доступна лише для нього.

5.2.3. Пункт меню “Користувачі”

Для перегляду списку зареєстрованих викладачів та студентів необхідно натиснути на кнопку “Опитування” (рисунок 5.6). Переглядати цей список може тільки адміністратор, тому ця кнопка доступна тільки для нього.

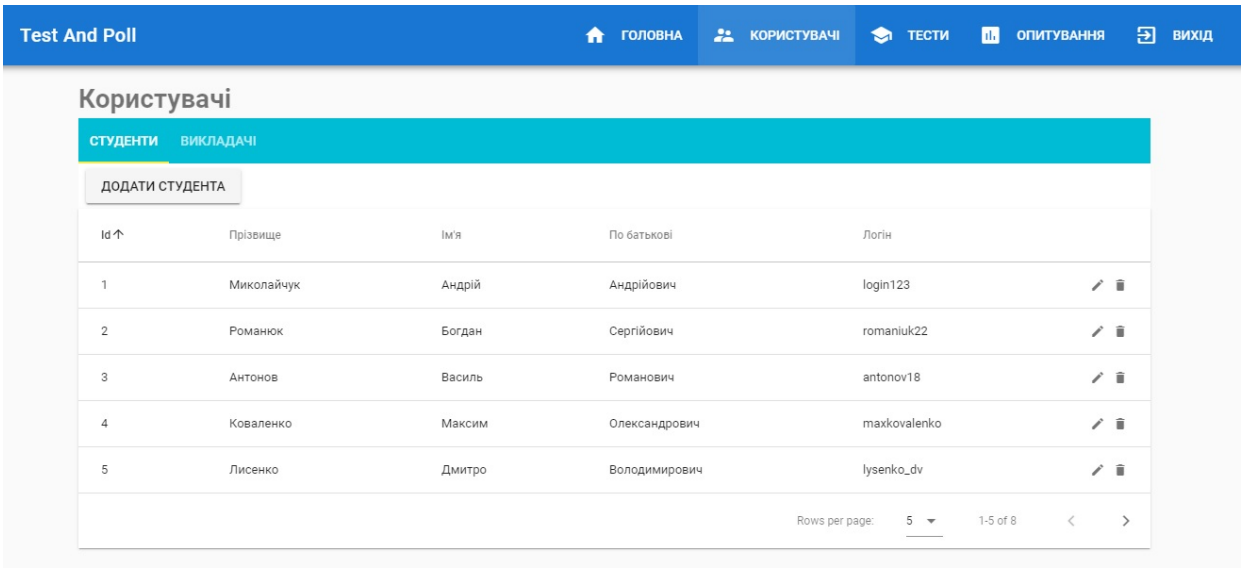

Рисунок 5.6 — Пункт меню “Користувачі”

Для редагування інформації про користувача адміністратор має натиснути навпроти рядка кнопку редагування, а для видалення облікового запису користувача — кнопку видалення.

5.2.4. Створення тесту або опитування

На рисунку 5.7 зображена форма створення тесту (опитування).

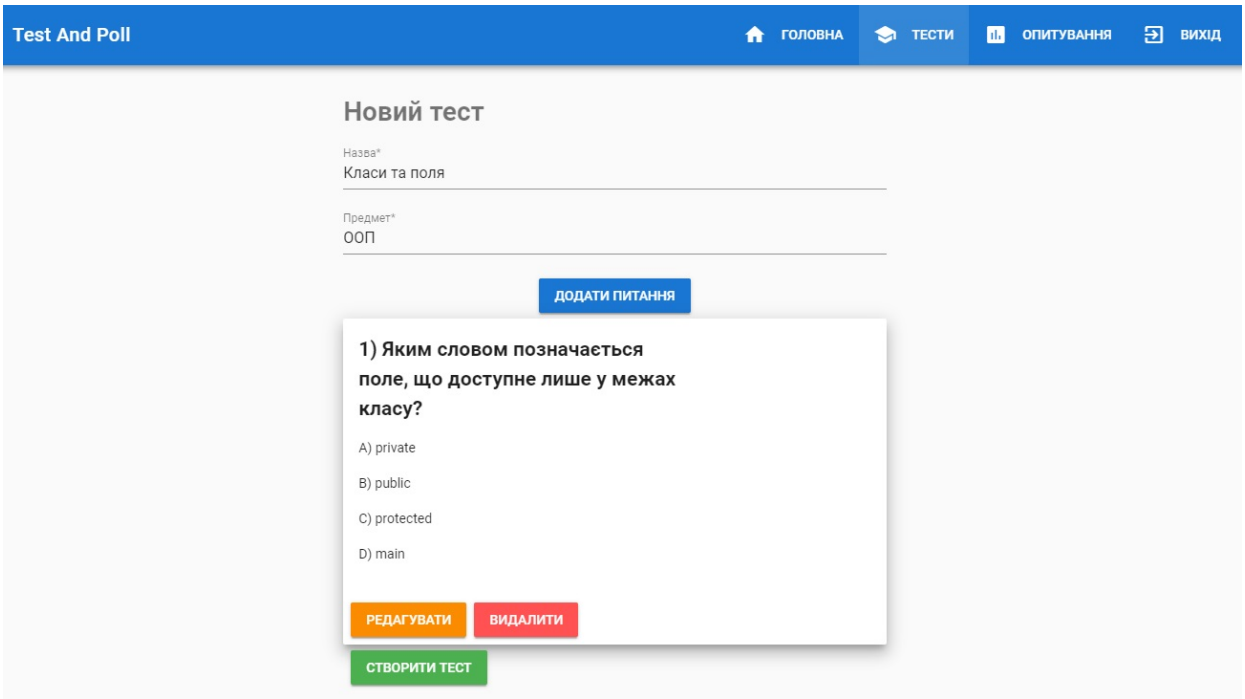

Рисунок 5.7 — Створення тесту (опитування)

Для того, щоб додати нове питання до тесту чи опитування, необхідно натиснути кнопку “Додати питання”.

Для редагування питання потрібно натиснути кнопку “Редагувати” на елементі списку створених питань, а для видалення — кнопку “Видалити”.

Якщо всі питання до тесту готові, треба натиснути на кнопку “Створити тест” (“Створити опитування”).

5.2.5. Проходження тесту або опитування

Для того, щоб знайти необхідний тест або опитування для проходження, студенту необхідно ввести всі необхідні дані до форми та обрати знайдений тест чи опитування.

На рисунку 5.8 зображене вікно проходження тесту (опитування).

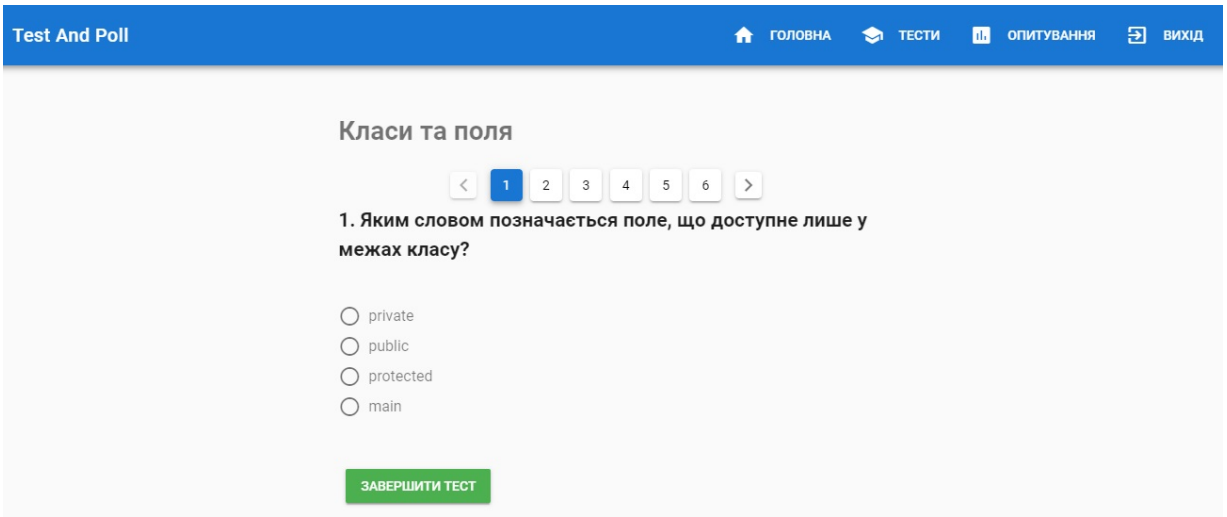

Рисунок 5.8 — Проходження тесту (опитування)

Для відповіді на питання достатньо обрати один з варіантів відповіді, натиснувши навпроти нього радіокнопку.

Щоб перейти на інше питання треба натиснути на номер питання.

Для завершення тесту необхідно натиснути кнопку “Завершити тест”.

ВИСНОВКИ

У ході аналізу існуючого програмного забезпечення організації та проведення тестів і опитувань у навчальному процесі було досліджено системи, які слугують для вирішення поставлених задач.

Розроблений програмний продукт дозволяє автоматизувати процеси тестування і опитування та допомагає це зробити дуже швидко.

Проведено огляд методів і засобів розробки програмної системи. Обґрунтовано вибір створення програмної системи, заснованої на веб-технологіях, а також побудованої за триланковою архітектурою. Це дає змогу підвищити гнучкість та зручність системи, як у розробці та супроводі, так і у використані.

За результатами виконання тестових завдань підтверджена коректність отриманих результатів, отже система відповідає поставленим вимогам.

Користувачами системи можуть бути викладачі, які створюють тести чи опитування, студенти, які їх проходять, а також адміністратори системи, які можуть змінювати інформацію про викладачів та студентів. Програмне забезпечення може бути використано на будь-якій операційній системі, на якій встановлено браузер, який підтримує останні веб-стандарти, а також яка має постійний доступ до інтернету.

Отже, практика покращила знання різноманітних технологій, що використовуються під час розробки програмного забезпечення. Також було створено декілька прототипів програмного забезпечення, які вирішували різноманітні аспекти поставленої задачі, а також які лягли в основу розробленого програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Биков В.Ю., Лапінський В.В. Методологічні та методичні основи створення і використовування електронних засобів навчального призначення // Комп'ютер у школі та сім'ї. — №3 . — 2012. С. 3-6.. 3.

2. Майоров А. Н. Теория и практика создания тестов для системы образования / Майоров А. Н. — М.: Народное образование, 2000. — 352 с.

3. Адаменко О.В., Духовна М.М., Панченко Л.Ф. та ін. Тестові завдання для контролю знань в курсі "Обчислювальна техніка і технічні засоби навчання": Навч.-метод, посібник / За ред. Т.О. Козлакової. — К.: ВІПОЛ, 1996.-84 с.

4. Алексейчук І.С. Про технологію створення системи тестування / І.С. Алексейчук // Нові технології навчання: Науково-методичний збірник. — К.: НМЦВД, 2000. — С.43-92.

5. Тетьякова Н. В. Оценка качества работы преподавателя на основе методики многомерного анализа его деятельности / Н. В. Третьякова // Научно-теоретический журнал "Ученые записки университета имени П. Ф. Лесгафта". — 2011. — №11(81). — С. 151-155.

6. Васильева, Е. Ю. Подходы к оценке качества деятельности преподавателя вуза // Университетское управление: практика и анализ. — 2006. — № 2 (11). — С. 74-78.

7. What is REST? [Електронний ресурс]. — Режим доступу: http://www.restapitutorial.com/lessons/whatisrest.html.

8. Markus Egger — MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF [Електронний ресурс]. — 2012. — Режим доступу: https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf.

9. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model. / К. Джонс — Перше, 2005. — 368 с.

10. Anthony Gore — Full-Stack Vue.js 2 and Laravel 5 [Електронний ресурс]. — 2015. — Режим доступу: https://bit.ly/2OEODzR.

11. Martin Fowler — GUI Architectures. Часть 1 [Електронний ресурс]. — 2009. — Режим доступу: https://bit.ly/2CvCk1e.

12. Vue.js Material Component Framework — Vuetify.js [Електронний ресурс]. — 2016. — Режим доступу: https://vuetifyjs.com.

13. Дмитрий Котеров, Алексей Костарев PHP. В подлиннике. / Д. Котеров, А. Костарев — Спб.: «БХВ-Петербург», 2005. — 1120 с.

14. Колисниченко Д. Н. Самоучитель PHP 5 / Д. Н. Колисниченко — СПб.: Наука и Техника, 2007. — 640 с.

15. Кузнецов Максим, Симдянов Игорь PHP 5/6. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2009. — 1024 с.

16. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на PHP. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2007. — 608 с.

17. Л. Аткінсон, З. Сураскін. PHP5. Бібліотека професіоналу. / Л. Аткінсон, З. Сураскін — М. : «Вільямс», 2006 — 543 с.

18. Скотт Хокінс. Адміністрування веб-сервера Apache і керівництво по електронній комерції. / С. Хокінс. — М.: «Вільямс», 2001. — 336 с.

19. Хокинс С. Администрирование Web-сервера Apache / С. Хокинс — М.: Вильямс, 2001. — 336 с.

20. Болье А. — Learning SQL [Електронний ресурс]. — 2005. — Режим доступу: http://shop.oreilly.com/product/9780596007270.do.

21. Роберт Шелдон, Джоффрей Мойе MySQL: базовый курс Beginning MySQL. / Р. Шелдон, Д. Мойе — М.: «Диалектика» 2007. — 880 с.

22. MySQL. Довідник. MySQL AB. — М: «Вільямс», 2006 — 521 с.

23. Кузнецов Максим, Симдянов Игорь MySQL на примерах. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2008. — 952 с.

24. Леон А. Г. PHP 5. Библиотека профессионала / А. Г. Леон — М. : Вильямс, 2006. — 944 с.

ДОДАТОК А

Web-ресурс для організації та проведення тестів і опитувань у навчальному процесі

Специфікація

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ-51182

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ51182	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ51182 12-1	TestList.vue	Модуль перегляду тестів
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ51182 12-2	PollList.vue	Модуль перегляду опитувань
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ51182 12-3	NewTest.vue	Модуль створення тесту
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ51182 12-4	SearchTest.vue	Модуль пошуку тесту для проходження
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ51182 12-5	api.php	Модуль керування запитами до бази даних

ДОДАТОК Б

Web-ресурс для організації та проведення тестів і опитувань у навчальному процесі

Текст програми

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ-51182

Аркушів 2

Київ 2019

Програмний модуль перегляду тестів

```
<template>
    <v-container>
        <v-layout row>
            <v-flex xs12 sm6 offset-sm3>
                <h1 class="text--secondary mb3">Тести</h1>
                <v-btn
                    :to="'/tests/new'"
                >Створити тест</v-btn>
                <v-btn class="success" :to="'/tests/search'">Пройти тест</v-btn>
                <v-card
                    class="elevation-10 mb2"
                    v-for="(test, i) in tests"
                    :key="i"
                >
                    <v-layout>
                        <v-flex xs8>
                            <v-card-text>
                                <h2 class="text--primary">{{ test.name }}</h2>
                                <p>Предмет: {{ test.subject }}</p>
                                <p>Автор: {{ test.author }}</p>
                                <p>Кількість питань: {{ test.amountQuestions }}</p>
                            </v-card-text>
                            <v-card-actions>
                                <v-spacer></v-spacer>
                                <v-btn
                                    class="info"
                                    :to="'/test/' + test.id + '/result'"
                                >Результати</v-btn>
                            </v-card-actions>
                        </v-flex>
                    </v-layout>
                </v-card>
            </v-flex>
        </v-layout>
    </v-container>
</template>

<script>
export default {
    data() {
        return {}
    },

    computed: {
        tests() {
            return this.$store.getters.tests
        }
    }
}
</script>
```

Програмний модуль перегляду опитувань

```
<template>
    <v-container>
        <v-layout row>
            <v-flex xs12 sm6 offset-sm3>
                <h1 class="text--secondary mb3">Опитування</h1>
                <v-btn
                    :to="'/polls/new'"
                >Створити опитування</v-btn>
                <v-btn class="success">Пройти опитування</v-btn>
                <v-card
                    class="elevation-10 mb2"
                    v-for="(poll, i) in polls"
                    :key="i"
                >
                    <v-layout>
                        <v-flex xs8>
                            <v-card-text>
                                <h2 class="text--primary">{{ poll.title }}</h2>
                                <p>Предмет: {{ poll.subject }}</p>
                                <p>Автор: {{ poll.author }}</p>
```

```
                            <p>Кількість питань: {{ poll.amountQuestions }}</p>
                        </v-card-text>
                        <v-card-actions>
                            <v-spacer></v-spacer>
                            <v-btn
                                class="info"
                                :to="'/poll/' + poll.id"
                            >Результати</v-btn>
                        </v-card-actions>
                    </v-flex>
                </v-layout>
            </v-card>
        </v-flex>
    </v-layout>
</v-container>
</template>

<script>
    export default {
        data() {
            return {}
        }

        computed: {
            polls() {
                return this.$store.getters.polls
            }
        }
    }
</script>
```

Програмний модуль створення тесту

```
<template>
    <v-container>
        <v-layout row v-if="!loading">
            <v-flex xs12 sm6 offset-sm3>
                <h1 class="text--secondary mb-3">Новий тест</h1>
                <v-form
                    ref="form"
                    v-model="valid">
                    <v-text-field
                        name="title"
                        label="Назва*"
                        type="text"
                        required
                        :rules="[v => !!v || 'Це поле обов\'язкове для введення']"
                        v-model="title"></v-text-field>
                    <v-text-field
                        name="amountQuestions"
                        label="Кількість питань*"
                        type="number"
                        min="1"
                        step="1"
                        required
                        :rules="[v => !!v || 'Це поле обов\'язкове для введення']"
                        v-model="amountQuestions"></v-text-field>
                        <EditTestQuestion :question="question"/>
                </v-form>
                <v-layout>
                    <v-flex xs12>
                            <v-card
                                class="elevation-10 mb2"
                                v-for="(question, i) in questions"
                                :key="i"
                            >
                                <v-layout>
                                    <v-flex xs8>
                                        <v-card-text>
                                            <h2>{{i + 1}}) {{question.name}}</h2>
                                            <p></p>
                                            <p>A) {{question.variantA}}</p>
                                            <p>B) {{question.variantB}}</p>
                                            <p>C) {{question.variantC}}</p>
                                            <p>D) {{question.variantD}}</p>
```

```
                                    </v-card-text>
                                    <v-card-actions>
                                        <v-btn class="warning">Редагувати</v-btn>
                                        <v-btn class="error">Видалити</v-btn>
                                    </v-card-actions>
                                </v-flex>
                            </v-layout>
                        </v-card>
                        <v-spacer></v-spacer>
                        <v-btn
                        :disabled="!valid"
                        class="success"
                        @click="createTest">Створити тест</v-btn>
                    </v-flex>
                </v-layout>
            </v-flex>
        </v-layout>
        <v-layout v-else>
            <v-progress-circular :value="100"></v-progress-circular>
        </v-layout>
    </v-container>
</template>

<script>
    import EditTestQuestion from "./TestQuestions/EditTestQuestion"
    import axios from "axios"

    export default {
        data() {
            return {
                title: '',
                amountQuestions: '1',
                select: null,
                subjects: [
                    'Item1',
                    'Item2'
                ],
                valid: false,

                questions: []
            }
        },
        components: {
            EditTestQuestion
        },

        computed: {
            loading() {
                return this.$store.getters.loading
            }
        },

        mounted() {
            axios("http://testandpoll.ua/api.php?action=read&obj=questions")
                .then(response => {
                    this.questions = response.data;
                })
                .catch(error => {
                })
        },

        methods: {
            createTest() {
                if(this.$refs.form.validate()) {
                    const test = {
                        title: this.title,
                        select: this.select,
                        amountQuestions: this.amountQuestions
                    }
                }
            },
        }
    }
</script>
```

Програмний модуль пошуку тесту для проходження

```
<template>
    <v-container>
        <v-layout row v-if="!loading">
            <v-flex xs12 sm6 offset-sm3>
                <h1 class="text--secondary mb-3">Новий тест</h1>
                <v-form
                    ref="form"
                    v-model="valid">
                    <v-text-field
                        name="title"
                        label="Викладач*"
                        type="text"
                        required
                        :rules="[v => !!v || 'Це поле обов\'язкове для введення']"
                        v-model="title"></v-text-field>
                    <v-text-field
                        name="amountQuestions"
                        label="Предмет*"
                        type="text"
                        required
                        :rules="[v => !!v || 'Це поле обов\'язкове для введення']"
                        v-model="amountQuestions"></v-text-field>
                </v-form>
                <v-card
                    class="elevation-10 mb2"
                    v-for="(test, i) in tests"
                    :key="i"
                >
                    <v-layout>
                        <v-flex xs8>
                            <v-card-text>
                                <h2>Name</h2>
                                <p>Topic</p>
                                <p>Author</p>
                                <p>Amount of Questions</p>
                            </v-card-text>
                        </v-flex>
                    </v-layout>
                </v-card>
            </v-flex>
        </v-layout>
        <v-layout v-else>
            <v-progress-circular :value="100"></v-progress-circular>
        </v-layout>
    </v-container>
</template>

<script>
    export default {
```

```
        data() {
            return {
                title: '',
                amountQuestions: '',
                select: null,
                subjects: [
                    'Item1',
                    'Item2'
                ],
                valid: false
            }
        },
        components: {
        },

        computed: {
            loading() {
                return this.$store.getters.loading
            },

            tests() {
                return this.$store.getters.tests
            }
        },

        methods: {
            createTest() {
                if(this.$refs.form.validate()) {
                    const test = {
                        title: this.title,
                        select: this.select,
                        amountQuestions: this.amountQuestions
                    }
                }
            },
        }
    }
</script>
```

Програмний модуль керування запитами до бази даних

```
<?php
    $db = mysql_connect('localhost', 'root', '') or die('');
    mysql_select_db('test_and_poll') or die('');
    $res = array('error', false);

    $action = 'read';
    $obj = 'tests';
    if(isset($_GET['action'])) {
        $action = $_GET['action'];
    }
    if(isset($_GET['obj'])) {
        $obj = $_GET['obj'];
    }

    if ($action == 'read' && $obj == 'tests') {
```

```
    $result = mysql_query("SELECT * FROM `tests`");
    $tests = array();

    while ($row = mysql_fetch_assoc($result)){
        array_push($users, $row);
    }
    $res['tests'] = $tests;
    echo json_encode($res['tests']);
}

if ($action == 'read' && $obj == 'polls') {
    $result = mysql_query("SELECT * FROM `polls`");
    $polls = array();

    while ($row = mysql_fetch_assoc($result)){
        array_push($users, $row);
    }
    $res['polls'] = $polls;
    echo json_encode($res['polls']);
}

if ($action == 'create' && $obj == 'tests') {
    $name = $_POST['name'];
    $subject = $_POST['subject'];

    var_dump($_POST);

    $result = mysql_query("INSERT INTO `tests`(`name`, `subject`) VALUES ('$name', '$subject') ");
    if ($result) {
        $res['message'] = "Test Added successfully";
        $res['name'] = $name;
        $res['subject'] = $subject;
    } else{
        $res['error'] = true;
        $res['message'] = "Insert Test fail";
    }
    echo json_encode($res);
}

if ($action == 'create' && $obj == 'polls') {
    $name = $_POST['name'];
    $subject = $_POST['subject'];

    var_dump($_POST);

    $result = mysql_query("INSERT INTO `polls`(`name`, `subject`) VALUES ('$name', '$subject') ");
    if ($result) {
        $res['message'] = "Poll Added successfully";
        $res['name'] = $name;
        $res['subject'] = $subject;
    } else{
        $res['error'] = true;
        $res['message'] = "Insert Poll fail";
    }
    echo json_encode($res);
}

if ($action == 'search' && $obj == 'tests') {
    $name = $_POST['name'];
    $subject = $_POST['password'];

    $result = mysql_query("SELECT * FROM `tests` WHERE `name` = '$name' AND `subject` = '$subject'");
    if ($result) {
        $result = [
            'name' => $result['name'],
            'subject' => $result['subject']
        ];
        $res['message'] =  "Test Search success";
        $res_int = mysql_numrows($result);
    } else{
        $res['error'] = true;
        $res['message'] = "Test Search failed";
    }
    echo json_encode($result);
}
```

```
    if ($action == 'search' && $obj == 'polls') {
        $name = $_POST['name'];
        $subject = $_POST['password'];

        $result = mysql_query("SELECT * FROM `polls` WHERE `name` = '$name' AND `subject` = '$subject'");
        if ($result) {
            $result = [
                'name' => $result['name'],
                'subject' => $result['subject']
            ];
            $res['message'] =  "Poll Search success";
            $res_int = mysql_numrows($result);
        } else{
            $res['error'] = true;
            $res['message'] = "Poll Search failed";
        }
        echo json_encode($result);
    }
?>
```

ДОДАТОК В

Web-ресурс для організації та проведення тестів і опитувань у навчальному процесі

Опис програми

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТІ-51182

Аркушів 2

Київ 2019

АНОТАЦІЯ

Даний додаток містить опис програмних модулів, розроблених для управління даними про тести і опитування. Створені програмні системи виконують такі завдання:

— перегляд результатів тестів і опитувань;

— створення нових тестів і опитувань;

— пошук нового тесту або опитування для його проходження.

Програмні модулі отримують дані через елементи керування, розташовані на формі.

При розробці програмних модулів використовувались мови JavaScript та PHP, із використанням прикладного програмного інтерфейсу API, веб-серверу Apache, системи керування базами даних MySQL та технології AJAX.

ЗМІСТ

ЗАГАЛЬНІ ВІДОМОСТІ

У цьому додатку міститься опис програмних модулів для управління тестами і опитуваннями. У додатку Б міститься програмний код головних модулів розроблюваних програмних систем.

Програмні модулі працюють у будь-якій операційній системі і потребують встановленого на ПК веб-серверу Apache та серверу бази даних MySQL.

При розробці програмних модулів використовувались мови JavaScript та PHP, із використанням прикладного програмного інтерфейсу API, веб-серверу Apache, системи керування базами даних MySQL та технології AJAX.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблені компоненти виконують завдання моделювання гідроакустичного процесу, а саме обчислюють швидкість звуку в солоній воді за допомогою формули Вільсона.

Також розроблені додатки можуть використовуватися в якості учбових матеріалів при опрацюванні технологій взаємодії застосунків C# та MATLAB.

Функціональні обмеження на використання компонентів полягають лише в діапазоні введення даних для обчислення швидкості.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Для реалізації керування тестами і опитуваннями знадобилось підключення до серверу бази даних за допомогою AJAX технології, яка дозволяє AJAX через її можливість використовувати API-запит веб-клієнта (браузера) до веб-сервера за протоколом HTTP у фоновому режимі.

В PHP були виконані SQL-запити для виконання операцій з базою даних.

При запуску модулів спочатку з’являється вікно виведення результатів про тести і опитування. Для створення чи проходження нового тесту чи опитування відбувається перехід на форму для введення даних, потім дані передаються до бази даних, обробляються, і кінцевий результат повертається назад на форму.

ТЕХНІЧНІ ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ

Для забезпечення повноцінної роботи та досягнення високої ефективності роботи створених програмних модулів для керування тестами і опитуваннями було обрано Visual Studio Code, яка показала себе надійною та гнучкою середою розробки програм.

Розроблені додатки працюють у будь-якій операційній системі і потребують встановленого на ПК веб-серверу Apache та серверу бази даних.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Оскільки застосунок працює з використанням веб-технологій, він не потребує інсталяції. Проте, для використання необхідний веб-браузер, який підтримує актуальні веб-стандарти. Також необхідно мати стабільний доступ до інтернету з швидкістю не менше ніж 50 мбіт за секунду. Для використання можливостей програми з проведення тестів і опитувань необхідно дати клієнтському додатку відповідний дозвіл.

Після запуску користувач отримає доступ до вікна керування тестами і опитуваннями, звідки може виконувати необхідні дії.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для розроблених програмних модулів є інформація, яка зчитується з textBox.

Вхідні дані мають текстовий вигляд.

Вихідними даними можуть мати вигляд цілого числа або тексту, які показуються у вікні перегляду результатів тесту чи опитування.

Вихідними даними є результати пройдених тестів і опитувань.